Package ‘GHAgricProductivityLab’

September 16, 2025

Type Package

Title Agricultural Productivity in Ghana

Version 0.0.0.9000

Author Francis Tsiboe [aut, cre] (<https://orcid.org/0000-0001-5984-1072>)

Maintainer Francis Tsiboe <ftsiboe@hotmail.com>

Contributor -

Reviewer -

Creator Francis Tsiboe

Description Provides tools and datasets for investigating the drivers of agricultural production shortfalls in Ghana.
It compiles research studies that examine farmer-specific and institutional factors to assess whether inefficiencies
arise from technical inefficiency, technology gaps, or both. The package offers empirical evidence to inform policy
discussions and interventions aimed at improving agricultural productivity,
particularly in contexts with limited access to modern technologies.

License GPL-3 + file LICENSE

URL https://github.com/you/GHAgricProductivityLab

BugReports https://github.com/you/GHAgricProductivityLab/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Depends R (>= 4.1.0)

Imports haven, data.table, MatchIt

Remotes github::dylan-turner25/rfcip

Suggests dplyr, knitr, crayon, tidyr, rmarkdown, testthat (>= 3.0.0)

LazyData true

Cite-us If you find it useful, please consider staring the repository and citing the following studies
- Tsiboe, F. and Turner, D. (2025). ``Incorporating buy-up price loss coverage into the United States
farm safety net.'' Applied Economic Perspectives and Policy.
- Tsiboe, F., et al. (2025). ``Risk reduction impacts of crop insurance in the United States.''

Applied Economic Perspectives and Policy.
- Gaku, S. and Tsiboe, F. (2024). Evaluation of alternative farm safety net program combination strategies. Agricultural Finance Review.

Contents

`data_Prep` *Prepare Data for Agricultural Productivity Analysis*

Description

Cleans and transforms a dataset by creating new variables, applying log transformations, converting selected variables to factors or characters, and recoding education levels. The function is designed to standardize inputs for further analysis of agricultural productivity.

Usage

```
data_Prep(data)
```

Arguments

`data`	A `data.frame` or `data.table` containing household- and farm-level variables such as weights, demographic information, and agricultural inputs.

Value

A cleaned and transformed `data.frame` or `data.table` with additional variables ready for analysis.

`draw_matched_samples` *Draw matched samples (stratified bootstrap + matching)*

Description

Performs stratified resampling at the `Surveyx, EaId` level (excluding EaIds listed for the current bootstrap `ID`) and computes adjusted sampling weights used for matching. Then fits a matching model per `m.specs[i, ]` using **MatchIt**, with exact matching on `Emch` and distance/link from `m.specs`.

Usage

```
draw_matched_samples(
  data,
  Emch,
  Scle,
  Fixd,
  m.specs,
  i,
  drawlist,
  verbose = FALSE
)
```

Arguments

`data`	A data.frame/data.table containing at least: `Surveyx`, `EaId`, `HhId`, `Mid`, `UID`, `Weight`, `Treat`, plus columns named in `Emch`, `Scle`, `Fixd`.
`Emch`	Character vector of covariate names used for exact matching.
`Scle`	Character vector of covariate names used in the distance model (scaling).
`Fixd`	Character vector of additional fixed covariates in the distance model.
`m.specs`	Data frame of matching specifications with columns `boot`, `method`, `distance`, and optionally `link`.
`i`	Integer index selecting the row of `m.specs` to use.
`drawlist`	Data frame where column `ID` identifies the bootstrap draw, and each remaining column corresponds to a `Survey` containing the sampled `EaId`.
`verbose`	Logical; if `TRUE`, prints the chosen matching method and timing. Default `FALSE`.

Details

Adjusted weights are computed as $pWeight = Weight \times (alloc/allocj)$, where `alloc` is the pre-exclusion sum of `Weight` by `Surveyx`, `EaId` and `allocj` is the post-exclusion sum.

Exact matching is performed on variables in `Emch`. The distance model formula is constructed as `Treat ~ Scle + Fixd`. When `distance == "glm"`, `m.order = "largest"` is used; otherwise `"closest"`.

Value

A list with:

`m.specs` The selected row from `m.specs`.

`m.out` The `matchit` object.

`md` Matched data: `Surveyx`, `EaId`, `HhId`, `Mid`, `UID`, `weights`, `pWeight`.

`df` The analysis data with adjusted weights: `Surveyx`, `EaId`, `HhId`, `Mid`, `UID`, `pWeight`.

draw_match_sample_specifications

Draw / Match Sample Specifications

Description

Generates (i) a draw list by sampling `EaId` within each unique `Survey` group and (ii) a grid of matching specifications for each bootstrap draw.

Usage

```
draw_match_sample_specifications(drawN, data, myseed = 3242025)
```

Arguments

`drawN` Integer. The number of draws to perform per `Survey`.

`data` A data.frame or data.table containing at least the columns `Survey` and `EaId`.

`myseed` Integer. Seed for random number generation (default `03242025`).

Details

Draw list: For each unique value of `Survey`, the function samples `drawN` `EaId` values with replacement and prepends a 0 row (ID = 0) for the baseline. The result is then spread to wide format with one column per `Survey`.

Matching specs: For each draw `ID`, creates a set of matching model specifications that include:

- Nearest neighbor with distances: `"euclidean"`, `"scaled_euclidean"`, `"mahalanobis"`, `"robust_mahalanobis"`
- Nearest neighbor with distance `"glm"` and links: `"logit"`, `"probit"`, `"cloglog"`, `"cauchit"`.

An `ARRAY` index is added for convenience.

Value

A list with two elements:

`m.specs` A data.frame of matching specifications with columns `boot`, `method`, `distance`, `link`, `ARRAY`.

`drawlist` A data.frame in wide format where each `Survey` is a column and rows correspond to draw `ID` (0:`drawN`).

Note

Requires that `data` contain `Survey` and `EaId`. `tidyr::spread()` is used for wide reshaping (consider `tidyr::pivot_wider()` in new code).

functional_forms *Generate Functional and Distribution Forms*

Description

Creates symbolic production function strings for stochastic frontier analysis (SFA), including Cobb–Douglas (CD) and Translog (TL) specifications, plus a set of candidate distributions for the inefficiency term.

Usage

```
functional_forms(nX = 5, trend = FALSE)
```

Arguments

nX	Integer. Number of input variables. Default is 5.
trend	Logical. If TRUE, adjusts the (optional) transcendental form (TP) by removing the last linear input term. Default is FALSE.

Details

The returned fxnforms list includes:

- CD: sum of log inputs (lnI1 + lnI2 + ...).
- TL: CD terms plus second-order and pairwise interaction terms (e.g., 0.5 * lnIi^2 and lnIi * lnIj for i < j).

Additional forms (Linear, Quadratic, Generalized, Transcendental) are shown in the code as commented examples; uncomment to include them. The trend argument only affects TP when that form is enabled.

distforms lists common inefficiency distributions (e.g., half-normal, truncated normal, exponential, lognormal, Weibull) and whether a scaling property is assumed for each.

Value

A list with:

fxnforms Named character strings of functional forms.

distforms Named list of inefficiency distributions with a scaling flag.

model_specifications *Build MSF Model Specifications*

Description

Creates a specification table for multi-stage frontier (MSF) analysis by combining production-function forms, distributional assumptions, technology variables, and disaggregation levels (pooled, crops, demographics).

Usage

```
model_specifications(
  data,
  distforms,
  fxnforms,
  TechVarlist,
  mainF = 2,
  mainD = 1,
  demographic_variables = c("Female", "Region", "Ecozon", "EduCat", "EduLevel", "AgeCat"),
  crop_list = c("Beans", "Cassava", "Cocoa", "Cocoyam", "Maize", "Millet", "Okra",
    "Palm", "Peanut", "Pepper", "Plantain", "Rice", "Sorghum", "Tomatoe", "Yam")
)
```

Arguments

`data`	data.frame/data.table providing values for `demographic_variables`.
`distforms`	Named list of distributions.
`fxnforms`	Named list of functional forms.
`TechVarlist`	Character vector of technology variables (first is default).
`mainF`	Integer index of preferred functional form (in fxnforms). Default 2.
`mainD`	Integer index of preferred distribution (in distforms). Default 1.
`demographic_variables`	Character vector of DATA column names for disaggregation.
`crop_list`	Character vector of crop names for disaggregation.

Value

A data.table with columns: disasg, level, TechVar, f, d.

Index